Concurrent Search of Mobile Users in Cellular Networks

Rung-Hung Gau, *Member, IEEE,* and Zygmunt J. Haas, *Senior Member, IEEE*

***Abstract*—In this paper, we propose to concurrently search for a number of mobile users in a wireless cellular network based on the probabilistic information about the locations of mobile users. The concurrent search approach guarantees that all k mobile users will be located within k time slots. It is shown that even in the worst case when mobile users appear equally in all the cells of the network, the concurrent search approach is able to reduce the average paging cost by 25%. More importantly, this is achieved without an increase in the worst case paging delay or in the worst case paging cost. Depending on the total number of mobile users to be located, total number of cells in the network, and the probabilistic information about the locations of mobile users, the reduction of the average paging cost due to the usage of the concurrent search approach ranges from 25% to 88%. The case in which perfect probabilistic information is unavailable is also studied.**

***Index Terms*—Cellular networks, concurrent search, probabilistic location information.**

I. Introduction

In a CELLULAR network, when a call to a mobile user arrives, a mobility management scheme is responsible for finding the current cell in which the mobile user resides. Typically, a mobility management scheme constitutes of a location update scheme and a paging scheme.

In the last decade, many location update schemes were proposed. Basically, these schemes were either movement-based [2], [13], timer-based [12], distance-based [8], [9], profile-based [21], state-based [22], or velocity-based [15]. Schemes that use a hybrid of the above strategies were also proposed [11], [19]. It was proven in [5] and [7] that distance-based schemes achieve better performance compared to movement-based schemes and timer-based schemes. Some proposed schemes [3], [20] suggested that a mobile should register its location only when it enters some predefined cells, referred to as reporting centers. Liang and Haas [23] proposed a predictive distance-based user tracking scheme. A novel information-theoretic approach for location update and derivation of probabilistic information about locations of mobile users was proposed in [24].

A location area is composed of a number of cells. Some researchers assumed that a mobile user sends a location update message to the system whenever it enters a new location area and concentrated on the design of an optimal location area. Kim and Lee [14] proposed an integer-programming model to find the optimal location area, which may take on an irregular shape. Abutaleb and Li [17] claimed that the problem of finding the optimal location area, when the size of the location area is constrained, is NP-hard. Other researchers assumed that location areas are given and focused on the decision problem of whether a mobile user should send a location update message when it enters a new location area. Das and Sen [18] proposed a genetic algorithm to decide whether a mobile should update its location when it enters a location area.

Recently, a convex optimization problem was formulated to minimize the costs of location update and paging in the movement-based location update scheme [32]. In [29], a continuous formulation for the problem of one-dimensional location area design is proposed to overcome the computational difficulty associated with the original combinatorial formulation. In [28], an improved probabilistic location update scheme was proposed. In [27], probabilistic paging is used for contention-free mobility management. A survey on location management schemes could be found in [31].

There is an intrinsic tradeoff between location update and paging. As the frequency of location update increases, the location uncertainty decreases and therefore the paging cost decreases. And on the contrary, when the frequency of location update decreases, both the location uncertainty and paging cost increase. It is possible to see paging as a more fundamental operation than location update. However, as pointed out in [24], "the majority of the research on location management has actually focused on update schemes, assuming some obvious version of the paging algorithm."

The concept of dividing a location area into paging zones was described in [1] and [4]. Lyberopoulos [10] proposed to page the cell that a mobile registered with most recently and then page all other cells in the location area if necessary. Rose and Yates [6] proved that given the probabilistic information about the position of a mobile, to minimize the average paging cost, the cells with the higher probabilities must be paged before the cells with the lower probabilities are paged. As described in [34], there are three types of methods to obtain the probabilistic information about the locations of mobile users: geographical computation, empirical data, and mathematical models. Krishnamachari, Gau, Wicker, and Haas [25] proposed an efficient algorithm to solve the problem of minimizing the average paging cost under the worst case paging delay constraint. Abutaleb and Li [16] showed that the problem of minimizing the average paging cost under the mean paging delay constraint can be solved in

Manuscript received January 4, 2002; revised June 14, 2002 and November 12, 2002; approved by IEEE/ACM Transactions on Networking Editor N. Vaidya. This work was supported in part by the Department of Defense Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-00-1-0564 and by the National Science Foundation under Grants ANI-9805094 and ANI-9704404.

R.-H. Gau is with the Institute of Communications Engineering, National Sun Yat-Sen University, Taiwan, R.O.C. (e-mail: runghung@mail.nsysu.edu.tw).

Z. J. Haas is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: haas@ece.cornell.edu).

Digital Object Identifier 10.1109/TNET.2003.822662

$O(2^n)$ time. A novel sequential paging strategy based on optimal search theory could be found in [33].

All the above works on paging focused on the problem of searching for a single mobile user and assumed that some straightforward strategy of searching for multiple mobile users, such as sequential search, is used. In this paper, we study the problem of searching for multiple mobile users concurrently. Furthermore, we propose an efficient and low-complexity algorithm to concurrently locate k mobile users within k time slots. Depending on the total number of cells in the network, the total number of mobile users to be located, and the probabilistic information about the locations of mobile users, the reduction of the average paging cost due to the usage of our proposed algorithm ranges from 25% to 88%. Our proposed scheme could be used in current cellular networks and next-generation personal communications networks.

The rest of this paper is organized as follows. Section II presents the mathematical formulation of the concurrent search problem. Section III shows that even in the worst case when mobile users appear equally in all the cells of the network, a concurrent search scheme is able to reduce the average paging cost by 25% without increasing the worst case paging delay or the average paging delay. Section IV contains the simple heuristic algorithm that is proposed to solve the concurrent search problem. Section V introduces the concept of the dynamic schedule that is used to further reduce the average paging cost and the conditional probability heuristic algorithm that is proposed to derive a dynamic schedule. Section VI shows our performance simulation results of the above algorithms. Section VII includes the discussions on the extensions of the proposed concurrent search algorithms. Conclusions are provided in Section VIII. Related proofs are given in the Appendix.

II. Concurrent Search Problem

Rose and Yates [6] proposed a sequential paging scheme to locate a single mobile user in the cellular network based on the probabilistic information of the location of the mobile user. The basic idea is to partition cells in the network into a number of paging zones and search for paging zones one by one. More precisely, the system pages the first paging zone in the first time slot. If the mobile user is in the first paging zone, the mobile user is located and the search is aborted. Otherwise, the system pages the second paging zone, and so on.

In this paper, we study the problem of concurrently locating a number of mobile users. Suppose there are n cells in the network and there are k mobile users to be located. A straightforward sequential paging scheme is to page all cells at time slot i to locate mobile user i, where $1 \leq i \leq k$. This scheme requires $k \cdot n$ paging messages. Given the probabilistic information about the locations of mobile users, it is possible to reduce the average number of required paging messages. Let $p(i,j)$ be the probability that mobile user i is in cell j, when a call to mobile user i arrives. To locate a mobile user, the system only searches for the cells in which the mobile user could be found with a nonzero probability. Therefore, without loss of essential generality, it is assumed that $p(i,j) > 0, \forall\, i \in \{1,2,\ldots,k\}$, $j \in \{1,2,\ldots,n\}$. Let P be a $k \times n$ probability matrix such that $[P]_{i,j} = p(i,j)$. A $k \times n$ matrix M is said to be a probability matrix if $\forall\, 1 \leq i \leq k$, $1 \leq j \leq n$, $[M]_{i,j} \in [0,1]$ and $\forall\, 1 \leq i \leq k$, $\sum_{j=1}^{n}[M]_{i,j} = 1$. As described in [34], there are three types of methods to obtain the mobile user location probabilities $p(i,j)$'s: geographical computation, empirical data, and mathematical models. In addition, a novel information-theoretical approach to derive the probability matrix P could be found in [24].

Consider the following simple example to illustrate the efficiency of a concurrent search scheme. Suppose that there are only two cells in the system and two users are to be located. Assume that the system has no prior knowledge about the location of the users; i.e., each user can reside in each one of the two cells with probability 0.5. Using sequential paging, the system would page and locate user 1 in the first time slot. Then, the system would page and locate user 2 in the second slot. Thus, total of four pages would be required.

On the other hand, if the system were to page in the first slot for user 1 in cell 1 and for user 2 in cell 2, there is 50% probability that any of the two users would be found in the first time slot. If a user is not found in the first time slot, the system would page for the user in the "other" cell in the second time slot. Thus, on the average, a user will be located with probability 0.5 in one time slot and with probability 0.5 in two time slots; i.e., on the average, 1.5 pages would be required per user, or 3 pages for the two users—a saving of 25% over the sequential paging.

In general, if there are n cells in the system and n users to be located, the above *concurrent* search scheme would require $n(n+1)/2$ total pages, while the sequential paging scheme requires $n \cdot n = n^2$ pages; i.e., a saving of 50% for large n! Of course, with some probabilistic knowledge of the mobiles' locations, this saving can be significantly larger.

Let $x(i,j)$ be the index of the time slot in which the base station in cell j is scheduled to page mobile user i. Let X be a $k \times n$ matrix such that $[X]_{i,j} = x(i,j)$. X is a schedule to search for mobile users. It is assumed that the complete schedule is determined at the beginning of the first time slot and remains unchanged. Suppose that all k mobile users must be located within d time slots. In order to assure that all k mobile users are located within d time slots, it is necessary that $1 \leq x(i,j) \leq d$, where $1 \leq i \leq k$, $1 \leq j \leq n$. The sth paging zone of mobile user i, where $1 \leq s \leq d$, $1 \leq i \leq k$, is denoted by $Z(i,s)$ and is defined to be $\{j | x(i,j) = s, 1 \leq j \leq n\}$. Moreover, the cardinality of $Z(i,s)$ is denoted by $|Z(i,s)|$, which is equivalent to the total number of elements in the set $Z(i,s)$. Let $I(condition)$ be the indicator function with value one if the condition is true and with value zero otherwise. Furthermore, let $\pi(i,s) = \sum_{j \in Z(i,s)} p(i,j) = \sum_{j=1}^{n} I(x(i,j) = s) \cdot p(i,j)$ be the probability that mobile user i is inside its sth paging zone, where $1 \leq i \leq k$ and $1 \leq s \leq d$. Let $m(i,s) = |Z(i,s)| = \sum_{j=1}^{n} I(x(i,j) = s)$, where $1 \leq i \leq k$ and $1 \leq s \leq d$.

A simple example is used to illustrate the above notations. Suppose there are $k = 2$ mobile users to be located within $d = 3$ time slots in a network composed of $n = 4$ cells. Assume that with probability 0.3, mobile user 1 is in cell 2. Therefore, $p(1,2) = 0.3$. In addition, it is assumed that $p(1,1) = 0.4$,

$p(1,3) = 0.2$, and $p(1,4) = 0.1$. Similarly, it is assumed that $p(2,1) = 0.3$, $p(2,2) = p(2,3) = 0.25$, and $p(2,4) = 0.2$. Therefore, the corresponding probability matrix P is as follows:

$$P = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.3 & 0.25 & 0.25 & 0.2 \end{bmatrix}.$$

Furthermore, assume that to locate mobile user one, the system pages cell 1 and cell 2 in time slot 1, and pages cell 3 in time slot 2, if the mobile has not been located at the end of time slot 1. If the mobile user has not been located at the end of time slot 2, the system pages cell 4 in time slot 3. Then, $Z(1,1) = \{1,2\}$, $Z(1,2) = \{3\}$, and $Z(1,3) = \{4\}$. Similarly, it is assumed that $Z(2,1) = \{3,4\}$, $Z(2,2) = \{1,2\}$, and $Z(2,3) = \emptyset$. Thus

$$X = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 2 & 2 & 1 & 1 \end{bmatrix}.$$

Moreover, $\pi(1,1) = 0.4 + 0.3 = 0.7$, $\pi(1,2) = 0.2$ and $\pi(1,3) = 0.1$. Similarly, $\pi(2,1) = 0.25 + 0.2 = 0.45$, $\pi(2,2) = 0.3 + 0.25 = 0.55$ and $\pi(2,3) = 0$. Clearly, $m(1,1) = 2$, $m(1,2) = 1$, $m(1,3) = 1$, $m(2,1) = m(2,2) = 2$ and $m(2,3) = 0$.

For mobile user i, with probability $\pi(i,s)$, the mobile user is located in the sth time slot and the total number of sent paging messages is $\sum_{\alpha=1}^{s} m(i,\alpha)$. Therefore, the average (expected) number of required paging messages to locate mobile user i is $\sum_{s=1}^{d}[\pi(i,s) \cdot \sum_{\alpha=1}^{s} m(i,\alpha)]$. Hence, the expected number of required paging messages to locate k mobile users is simply $\sum_{i=1}^{k}\sum_{s=1}^{d}[\pi(i,s)\cdot\sum_{\alpha=1}^{s} m(i,\alpha)]$. Given the probabilistic information P and the schedule of searching for mobile users X, the average paging cost is defined to be the expected number of required paging messages to locate mobile users and is denoted by $C(P,X)$. Then

$$\begin{aligned} C(P,X) &= \sum_{i=1}^{k}\sum_{s=1}^{d}\left[\pi(i,s)\cdot\sum_{\alpha=1}^{s} m(i,\alpha)\right] \\ &= \sum_{i=1}^{k}\sum_{s=1}^{d}\left\{\left[\sum_{j=1}^{n} I\left(x(i,j)=s\right)\cdot p(i,j)\right]\right. \\ &\qquad \left.\cdot\left[\sum_{\alpha=1}^{s}\sum_{j=1}^{n} I\left(x(i,j)=\alpha\right)\right]\right\}. \end{aligned} \tag{1}$$

There are many schedules to search for mobiles users and there are natural constraints that a valid schedule must satisfy. In this paper, it is assumed that in a single time slot, a base station is able to broadcast at most one paging message and a paging message is used to search for a specific mobile user. While a base station is allowed not to send any paging messages in a time slot, a base station is prohibited from sending a paging message to search for two or more mobile users in a time slot. Therefore, for every cell, mobile users are scheduled to be paged in distinct time slots. Namely, $\forall\, j, i_1, i_2$, where $1 \le j \le n$, $1 \le i_1, i_2 \le k$, if $i_1 \ne i_2$, then $x(i_1,j) \ne x(i_2,j)$.

A valid schedule also has to guarantee that all mobile users are located within d time slots. Therefore, for every cell, every mobile user has to be scheduled to be paged in the cell within d time slots. Instead of using the term "be paged", we use the term "scheduled to be paged" to emphasize that once a mobile user has been located, it is unnecessary to search for the mobile user in remaining unpaged cells.

Let $S(k,n,d)$ be the collection of valid schedules to locate k mobile users in a n-cells network within d time slots. Then

$$\begin{aligned} S(k,n,d) = \{X | x(i,j) \in \{1,2,\ldots,d\}, \quad &\forall\, i \in \{1,2,\ldots,k\}, \\ &j \in \{1,2,\ldots,n\}, \text{ and } \quad \forall\, j, i_1, i_2, \\ &\text{where } 1 \le j \le n,\ 1 \le i_1,\ i_2 \le k, \\ &\text{if } i_1 \ne i_2, \text{ then } x\left(i_1,j\right) \ne x(i_2,j)\}. \end{aligned} \tag{2}$$

A necessary condition for $S(k,n,d)$ to be a nonempty set is $d \ge k$. In this paper, only the nontrivial case in which $S(k,n,d) \ne \emptyset$ is studied.

The optimal average paging cost $C_d^*(P)$ and the optimal schedule X^* are defined as follows:

$$\begin{aligned} C_d^*(P) &= \min_{X \in S(k,n,d)} C(P,X) \\ \text{and } X^* &= \arg\min_{X \in S(k,n,d)} C(P,X). \end{aligned} \tag{3}$$

The concurrent search problem is to find an optimal schedule and the minimum average paging cost, given k, n, d, and P. In the above definition of a schedule, it is assumed that the complete schedule is determined based on the probability matrix P at the beginning of the first time slot and remains unchanged. Therefore, the above optimal schedule is an optimal static schedule. It is possible to further reduce the average paging cost by a dynamic schedule, which will be explained in more details later in this paper. In this paper, unless explicitly stated, an optimal schedule means an optimal static schedule.

When $d \ge k \cdot n$, the concurrent search problem is identical to a collection of k independent sequential paging problems. An optimal schedule is to search for mobile users one by one and for each mobile user i, sort the probabilities $p(i,j)$, where $1 \le j \le n$, in nonincreasing order and page cells one by one accordingly.

In this paper, we focus on the case in which $d = k$. In this case, $S(k,n,k) \ne \emptyset$, since there is a trivial solution in which all the base stations of the network page mobile user i in time slot i, where $1 \le i \le k$. When $d = k$, all k mobile users will be located within k time slots and therefore the worst case paging delay is k, which is identical to the worst case paging delay of the straightforward sequential paging scheme. The constraint that $\forall\, j$, i_1, i_2, where $1 \le j \le n$, $1 \le i_1$, $i_2 \le k$, if $i_1 \ne i_2$, then $x(i_1,j) \ne x(i_2,j)$ is essential for the concurrent search problem. Due to the constraint, the general concurrent search problem is not equivalent to a collection of independent sequential paging problems [6]. Consider the following counter example. Suppose there are two mobile users and two cells in the network. Furthermore, assume that $p(1,1) > p(1,2)$ and $p(2,1) > p(2,2)$. If the concurrent search problem is identical to a collection of independent sequential paging problems, due to the rule that the cells with higher probabilities must be paged first, to locate mobile user one, cell one has to be paged before cell 2 is paged. Thus, cell one has to be paged for mobile user one in time slot one. Similarly, to locate mobile user two, cell one has to be paged for mobile user two in time slot one. However, the base station in cell one is not allowed to simultaneously page two mobile users in the first time slot.

The concurrent search problem can be solved by the "brute force" search algorithm, which would simply search for the whole state space $S(k,n,k)$. The cardinality of $S(k,n,k)$ is denoted by $|S(k,n,k)|$ and is calculated as follows. For every fixed cell, two mobile users cannot be paged at the same time. On the other hand, in a valid schedule, for every fixed cell, the corresponding base station has to page all k mobile users in the first k time slots to satisfy the requirement that all k mobile users are located within k time slots. Therefore, for every fixed cell j, $x(1,j), x(2,j), \ldots, x(k,j)$ must be a permutation of the sequence $1, 2, \ldots, k$. There are totally $k!$ distinct permutations. Since a base station is able to page any mobile user independent of other base stations, the cardinality of $S(k,n,k)$ is equivalent to $(k!)^n$, where n is the total number of cells in the network. The high computational complexity of the algorithm even for small n and k makes the "brute force" search impractical.

A similar concurrent search problem to minimize the average paging delay is formulated as follows. Given k, n, d, a probability matrix P and a valid schedule X, let $D(P,X)$ be the associated average paging delay. Then

$$\begin{aligned} D(P,X) &= \frac{1}{k}\sum_{i=1}^{k}\sum_{s=1}^{d}[\pi(i,s)\cdot s] \\ &= \frac{1}{k}\sum_{i=1}^{k}\sum_{s=1}^{d}\left\{\left[\sum_{j=1}^{n}\left(I\left(x(i,j)=s\right)\cdot p(i,j)\right)\right]\cdot s\right\}. \end{aligned}$$

Furthermore, given k, n, d, and P, denote the optimal average paging delay by $D_d^*(P)$, where

$$D_d^*(P) = \min_{X\in S(k,n,d)} D(P,X).$$

III. Uniform Distributions

A. Average Paging Cost

In this section, we show that even when every mobile user resides equally in all the cells, which is the worst case for improvement with the concurrent search scheme, reduction of the average paging cost can still be achieved by concurrent search. In this case, solving independent sequential paging problems can solve the concurrent search problem.

Let $P_{U(k,n)}$ be a $k\times n$ matrix such that $[P_{U(k,n)}]_{i,j} = 1/n$, where $1 \le i \le k, 1 \le j \le n$. Given a $k\times n$ probability matrix P, let $C_{\text{seq}}(P)$ be the paging cost, when k mobile users in a n-cells network are searched one by one and all base stations simultaneously page mobile user i in time slot i, where $1 \le i \le k$. It is clear that $C_{\text{seq}}(P_{U(k,n)}) = k\cdot n$.

The case in which $k = 2$ and n is an even number is first studied. Recall that X is the collection of the $x(i,j)$'s, where $1 \le i \le 2$, $1 \le j \le n$. The following concurrent search schedule is selected:

$$\begin{aligned} x(1,j) &= 1, \quad \forall\, 1 \le j \le \frac{n}{2} \\ x(1,j) &= 2, \quad \forall\, \frac{n}{2}+1 \le j \le n \\ x(2,j) &= 2, \quad \forall\, 1 \le j \le \frac{n}{2} \\ x(2,j) &= 1, \quad \forall\, \frac{n}{2}+1 \le j \le n. \end{aligned}$$

It is straightforward to check that X is a valid schedule. Recall that $C_2^*(P)$ is the minimum average paging cost, when the probability matrix is P and all mobile users are located within two time slots. Then

$$\begin{aligned} C_2^*\left(P_{U(2,n)}\right) &\le C_2\left(P_{U(2,n)}, X\right) \\ &= 2\cdot\left[\left(\frac{1}{2}\right)\cdot\left(\frac{n}{2}\right) + \left(\frac{1}{2}\right)\cdot(n)\right] \\ &= \frac{3\cdot n}{2}. \end{aligned}$$

The normalized reduction of the average paging cost for a $k\times n$ probability matrix P is denoted by $r^*(k,n,P)$ and is defined as follows:

$$r^*(k,n,P) = \frac{C_{\text{seq}}(P) - C_k^*(P)}{C_{\text{seq}}(P)}.$$

Then

$$r^*\left(2,n,P_{U(2,n)}\right) \ge \frac{2\cdot n - \frac{3\cdot n}{2}}{2\cdot n} = \frac{1}{4}.$$

The above result demonstrates that even when every mobile user appears equally in all the cells of the network, which is considered to be the worst case, reduction in the average paging cost of 25% is achievable by concurrent search.

It turns out that the schedule chosen above is indeed an optimal schedule, which is proved as follows. Assume that k divides n. Recall that $m(i,s)$, where $1 \le i, s \le k$, is the total number of cells that belong to the sth paging zone of mobile user i. Let m be the collection of $m(i,s)$'s, where $1 \le i$, $s \le k$. Let $C(m)$ be the corresponding average paging cost. For $P_{U(k,n)}$, $\pi(i,s) = m(i,s)/n$ and therefore

$$C(m) = \sum_{i=1}^{k}\sum_{s=1}^{k}\left[\frac{m(i,s)}{n}\cdot\sum_{\alpha=1}^{s} m(i,\alpha)\right].$$

There are a number of constraints that m has to satisfy. First, by definition, $m(i,s) = |Z(i,s)|$, where $1 \le i$, $s \le k$, is an integer between zero and n. In addition, to assure that mobile user i is located within k time slots, $\sum_{s=1}^{k} m(i,s) = \sum_{s=1}^{k}|Z(i,s)| = n$. Moreover

$$\begin{aligned} \sum_{i=1}^{k} m(i,s) &= \sum_{i=1}^{k}\sum_{j=1}^{n} I\left(x(i,j)=s\right) \\ &= \sum_{j=1}^{n}\sum_{i=1}^{k} I\left(x(i,j)=s\right) \\ &= \sum_{j=1}^{n} 1 \\ &= n. \end{aligned}$$

Therefore, to find an optimal solution m and the minimum value of $C(m)$, we have to solve the following optimization problem:

$$
\begin{aligned}
&\min \sum_{i=1}^{k}\sum_{s=1}^{k}\left[\frac{m(i,s)}{n}\cdot\sum_{\alpha=1}^{s} m(i,\alpha)\right]\\
&\text{s.t.}\\
&\quad \sum_{s=1}^{k} m(i,s)=n, \quad \forall\ 1\le i\le k\\
&\quad \sum_{i=1}^{k} m(i,s)=n, \quad \forall\ 1\le s\le k\\
&\quad m(i,s)\in\{0,1,2,\ldots,n\}, \quad \forall\ 1\le i,\ s\le k.
\end{aligned}
$$

Let $m^*(i,s)=n/k, \forall\, 1\le i, s\le k$ and m^* be the collection of the $m^*(i,s)$'s. Since k divides n, n/k is an integer. Since $\sum_{s=1}^{k} m(i,s)=k\cdot(n/k)=n, \sum_{i=1}^{k} m(i,s)=k\cdot(n/k)=n$, and n/k is an integer between 0 and n, m^* is a feasible solution of the above optimization problem. Furthermore

$$
\begin{aligned}
C(m^*) &= \sum_{i=1}^{k}\sum_{s=1}^{k}\left[\frac{1}{k}\sum_{\alpha=1}^{s}\frac{n}{k}\right]\\
&= \sum_{i=1}^{k}\sum_{s=1}^{k}\left[\left(\frac{1}{k}\right)\cdot\left(\frac{n\cdot s}{k}\right)\right].
\end{aligned}
$$

Let C_i^* be the minimum average paging cost, when an optimal sequential paging scheme [6] is used to locate a single mobile user i within k time slots. In [25], it has been shown that

$$
\begin{aligned}
C_i^* &= \sum_{s=1}^{k}\left[\frac{1}{k}\sum_{\alpha=1}^{s}\frac{n}{k}\right]\\
&= \sum_{s=1}^{k}\left[\left(\frac{1}{k}\right)\cdot\left(\frac{n\cdot s}{k}\right)\right]
\end{aligned}
$$

Therefore, $C(m^*)=\sum_{i=1}^{k} C_i^*$.

Without the last set of constraints, $\sum_{i=1}^{k} m(i,s)=n$, $\forall\ 1\le s\le k$, the concurrent search problem is equivalent to a collection of independent optimal sequential paging problems. We now prove that if m is a solution of the above concurrent search problem, $C(m)\ge\sum_{i=1}^{k} C_i^*$. For each fixed i, where $i\in\{1,2,\ldots,k\}$, since $\sum_{s=1}^{k} m(i,s)=n$ and $m(i,s)\in\{0,1,2,\ldots,n\}$, $\forall\ s\in\{1,2,\ldots,k\}$, $m(i,1), m(i,2),\ldots,m(i,k)$ form a solution of the optimal sequential paging problem [6]. Hence, $\forall\ i\in\{1,2,\ldots,k\}$, $\sum_{s=1}^{k}(m(i,s)/n)\sum_{\alpha=1}^{s} m(i,\alpha)\ \ge\ C_i^*$ and therefore $C(m)\ge\sum_{i=1}^{k} C_i^*$. Then, $C(m^*)=\sum_{i=1}^{k} C_i^*\le C(m)$, and therefore, m^* is an optimal solution.

Lemma 1: When k divides n, an optimal schedule is $x(i,j)=i-1+\lceil j\cdot k/n\rceil-k\cdot I(i-1+\lceil j\cdot k/n\rceil>k)$, $\forall\ 1\le i\le k, 1\le j\le n$.

Proof: See the Appendix.

Lemma 2: When k divides n, $r^*(k,n,P_{U(k,n)})=(1/2)-(1/(2\cdot k))$.

Proof: See the Appendix.

B. Average Paging Delay

Recall that $m^*(i,s)=n/k, \forall\ 1\le i,\ s\le k$ and m^* is the collection of the $m^*(i,j)$'s. Then

$$
D\left(P_{U(k,n)}, m^*\right)=\frac{1}{k}\sum_{i=1}^{k}\sum_{s=1}^{k}\frac{s}{k}=\frac{k+1}{2}.
$$

When k mobile users are searched one by one, the average paging delay is equivalent to $(1/k)\sum_{i=1}^{k} i=(k+1)/2$, which is identical to $D(P_{U(k,n)}, m^*)$. Thus, the concurrent search scheme does not increase the average paging delay in the uniform user distribution case.

When k mobile users are searched one by one, the last mobile user is always located in time slot k. When a concurrent search scheme is used, in the worst case, it is possible that some mobile users might be located as late as time slot k. Therefore, the worst case paging delay of a concurrent search scheme is also k time slots. To sum up, when mobile users appear equally in all of the cells in the network, the concurrent search scheme is able to reduce the average paging cost without increasing the average paging delay or the worst case paging delay.

IV. Algorithms and Analytical Results

We propose the simple heuristic algorithm to solve the concurrent search problem $\min_{X\in S(k,n,k)} C(P,X)$. Although to the best of our knowledge, the "brute force" search algorithm is the only known algorithm that guarantees an optimal solution, its computational complexity is exponential, and therefore, it is not practical. The simple heuristic algorithm is a linear-time algorithm and it guarantees the normalized reduction of the average paging cost by at least 16.6%. Related analytical results are also included.

A. Simple Heuristic Algorithm

To evaluate the performance of a heuristic algorithm, we extend the previous definition of the normalized reduction of the average paging cost. For an arbitrary schedule $X\in S(k,n,k)$ and a $k\times n$ probability matrix P, the normalized reduction of the average paging cost is denoted by $r(k,n,P,X)$ and is defined to be $(C_{\text{seq}}(P)-C(P,X))/C_{\text{seq}}(P)$. We further define $r^*(k,n,P)=\max_{X\in S(k,n,k)} r(k,n,P,X)$.

The simple heuristic algorithm for $k=2$ is as follows. Let $\phi_1=\sum_{j=1}^{\lceil n/2\rceil} p(1,j)$ and $\phi_2=\sum_{j=1}^{\lceil n/2\rceil} p(2,j)$. If $\phi_1\ge\phi_2$, set $x(1,j)=1, x(2,j)=2, \forall\ 1\le j\le\lceil n/2\rceil$, and $x(1,j)=2$, $x(2,j)=1, \forall\ \lceil n/2\rceil+1\le j\le n$. Otherwise, set $x(1,j)=2$, $x(2,j)=1, \forall\ 1\le j\le\lceil n/2\rceil$, and $x(1,j)=1, x(2,j)=2$, $\forall\ \lceil n/2\rceil+1\le j\le n$.

Let X_s be the schedule derived by the simple heuristic algorithm. An upper bound of $C(P,X_s)$, when $k=2$, is derived as follows. If n is an odd number, $\lceil n/2 \rceil = (n+1)/2$. Define $h(n) = I$ (n is an odd number). When $\phi_1 \geq \phi_2$

$$\begin{aligned} C(P, X_s) &= \left[\phi_1 \cdot \left\lceil \frac{n}{2} \right\rceil + (1-\phi_1) \cdot n\right] \\ &\quad + \left[(1-\phi_2) \cdot \left(n - \left\lceil \frac{n}{2} \right\rceil\right) + \phi_2 \cdot n\right] \\ &= \left(2 \cdot n - \left\lceil \frac{n}{2} \right\rceil\right) - \phi_1 \cdot \left(n - \left\lceil \frac{n}{2} \right\rceil\right) + \phi_2 \cdot \left\lceil \frac{n}{2} \right\rceil \\ &= \frac{3n - h(n)}{2} - \phi_1 \cdot \left(\frac{n - h(n)}{2}\right) \\ &\quad + \phi_2 \cdot \left(\frac{n + h(n)}{2}\right) \\ &= \frac{3n + (2\phi_2 - 1) \cdot h(n)}{2} - (\phi_1 - \phi_2) \\ &\quad \cdot \left(\frac{n - h(n)}{2}\right) \\ &\leq \frac{3n + (2-1) \cdot h(n)}{2} - 0 \\ &= \frac{3n + h(n)}{2}. \end{aligned}$$

Similarly, when $\phi_1 \leq \phi_2$, it can be proved that $C(P, X_s) \leq (3n + h(n))/2$. Therefore

$$\begin{aligned} r(2, n, P, X_s) &\geq 1 - \frac{\frac{3n+h(n)}{2}}{2n} \\ &= \frac{1}{4} - \frac{h(n)}{4n}. \end{aligned}$$

The above results are summarized in the following lemma.

Lemma 3: $r(2, n, P, X_s) \geq (1/4) - (h(n)/4n)$.

Particularly, when there are two mobile users and the total number of cells is an even number, according to the above lemma, the normalized reduction of the average paging cost due to the simple heuristic algorithm is at least 25%.

The simple heuristic algorithm is also applicable when there are $k \geq 3$ mobile users to be located. When k is an even number, for every pair of mobile users, the above procedure is used to locate the mobile users. Namely, the simple heuristic algorithm searches for mobile user 1 and mobile user 2 in the first two time slots, searches for mobile user 3 and mobile user 4 in time slot 3 and time slot 4, and so on. In general, the algorithm pairs and searches for the mobile user $(2 \cdot i - 1)$ and the mobile user $(2 \cdot i)$ at time slot $(2 \cdot i - 1)$ and time slot $(2 \cdot i)$. It can be shown that $r(k, n, P, X_s) \geq (1/4) - (h(n)/4n)$.

When k is an odd number and $k = 2 \cdot k' + 1$, the simple heuristic algorithm uses the above procedure to locate the first $2 \cdot k'$ mobile users in the first $2 \cdot k'$ time slots and then all base stations simultaneously page the last mobile user in the kth time slot. In this case, $r(k, n, P, X_s) \geq (1/4) - (1/4k) - (1 - (1/k)) \cdot (h(n)/4n)$. When $k = 3$, the above lower bound is about 16.6%. As k and n increase, the lower bound approaches 25%, which is identical to the limit of the lower bound when k is even.

Let $l_s(k, n) = (1/4) - (h(n)/4n)$, when k is an even number, and $l_s(k, n) = (1/4) - (1/4k) - (1 - (1/k)) \cdot (h(n)/4n)$, when k is an odd number. Since the simple heuristic algorithm derives a suboptimal solution of the concurrent search problem, $r^*(k, n, P) \geq l_s(k, n)$. Namely, $l_s(k, n)$ is a lower bound of the normalized reduction of the average paging cost, when the "brute force" algorithm is used to find an optimal schedule. Note that $l_s(k, n)$ is independent of the probability matrix P. Compared to searching for mobile users one by one, an optimal concurrent search always reduces the normalized average paging cost by at least 16.6%. When the total number of mobile users is an even number, the above reduction is at least 25%.

The simple heuristic algorithm takes $O((k \cdot n)/2)$ additions and $O(k/2)$ comparisons to obtain a schedule. Therefore, its computational complexity is $O(k \cdot n)$, which is practical for implementation with today's technology and cellular system parameters.

B. Analytical Results of the Average Paging Cost

We first prove that when there are two mobile users, the collection of two uniform probability density functions is the worst case for minimizing the average paging cost.

Lemma 4: $C_2^*(P_{U(2,n)}) \geq C_2^*(P)$, where n is an even number and P is an arbitrary $2 \times n$ probability matrix.

Proof: See the Appendix.

However, unlike in the sequential paging problem, the collection of uniform distributions is not the unique worst case in the concurrent search problem. One example is shown by the following lemma.

Lemma 5: If $n = k = 2$ and $p(1, j) = p(2, j), \forall\, 1 \leq j \leq n$, then $C_2^*(P) = C_2^*(P_{U(2,n)})$.

Proof: See the Appendix.

Definition: A matrix is said to be row-identical if all rows are identical.

In the sequential paging problem, if the cells with lower probabilities are paged before the cells with higher probabilities are paged, the average paging cost associated with a nonuniform distribution could be greater than that associated with the uniform distribution. This suggests that a row-identical probability matrix is a bad case for the concurrent search problem, since for some mobile user, some cells with lower probabilities have to be paged before some other cells with higher probabilities are paged. However, the following lemma shows that the average paging cost associated with a $k \times n$ row-identical probability matrix is no greater than that associated with $P_{U(k,n)}$.

Lemma 6: If k divides n and P_I is a $k \times n$ row-identical probability matrix, $C_k^*(P_I) \leq C_k^*(P_{U(k,n)})$.

Proof: See the Appendix.

C. Analytical Results of the Average Paging Delay

The following lemma states that when there are two mobile users, the collection of two uniform probability density functions is the worst case for minimizing the average paging delay.

Lemma 7: $D_2^*(P_{U(2,n)}) \geq D_2^*(P)$, where P is an arbitrary $2 \times n$ probability matrix.

Proof: See the Appendix.

It is shown in the second step of the above proof that when there are two mobile users that reside equally likely in all the cells in the network, the average paging delay is always 3/2, regardless of the schedule. Following the steps of the proof, it is easy to show that when there are k mobile users that reside equally likely in all the cells in the network, the average paging delay is always $(k+1)/2$.

The following lemma shows that when $k = 2$, $P_{U(k,n)}$ is not the unique worst case for the problem of minimizing the average delay.

Lemma 8: If $k = 2$ and $p(1,j) = p(2,j), \forall\, 1 \leq j \leq n$, then $D_2^*(P) = D_2^*(P_{U(2,n)}), \forall\, n \geq 2$.

Proof: See the Appendix.

The following lemma states that the average paging delay associated with a $k \times n$ row-identical probability matrix is no greater than that associated with $P_{U(k,n)}$.

Lemma 9: If k divides n and P_I is a $k \times n$ row-identical probability matrix, then $D_k^*(P_I) \leq D_k^*(P_{U(k,n)})$.

Proof: See the Appendix.

V. On the Dynamic Schedule

To further reduce the average paging cost, we propose the conditional probability heuristic algorithm to obtain a dynamic schedule. Unlike a static schedule, a dynamic schedule is not fully determined at the beginning of the first time slot. Instead, a dynamic schedule is obtained based on the paging results of the previous time slots.

A. Conditional Probability Heuristic Algorithm

We first define some terms for the conditional probability heuristic algorithm. The variable $s(i,t)$ represents the state of the ith mobile user at the beginning of time slot t. If mobile user i has not been located at the beginning of time slot t, $s(i,t) = 0$. Otherwise, $s(i,t) = 1$. Since none of the mobile users are located at the beginning of time slot 1, $s(i,1) = 0$, where $1 \leq i \leq k$. The variable $A(t)$ stores the up-to-date probabilistic information about the locations of mobile users. Initially, $A(1) = P$, where P is the probability matrix defined earlier. At the end of time slot t, $A(t+1)$ is calculated based on $A(t)$ and the results of paging mobile users in time slot t. The variable $\phi_j(t)$ is the index of the mobile user that is paged by the base station of cell j in time slot t.

The conditional probability heuristic algorithm is presented as follows:

```
Step 0) Initially, $t = 1$, $A(1) = P$ and
        $s(i,1) = 0$, $\forall\ 1 \leq i \leq k$.
Step 1) At the beginning of time slot $t$,
        for each $j$, where $1 \leq j \leq n$,
        choose $\phi_j(t)$ such that $[A(t)]_{\phi_j(t),j} =$
        $\max_{1 \leq i \leq k}[A(t)]_{i,j}$. Then, for each $j$,
        where $1 \leq j \leq n$, if $\max_{1 \leq i \leq k}[A(t)]_{i,j} >$
        $0$, the base station of the $j$th cell
        pages mobile user $\phi_j(t)$.
Step 2) At the end of time slot $t$, calcu-
        late $s(i,t+1)$'s as follows:
          If mobile user $i$ is located in
          time slot $t$, then $s(i,t+1) = 1$.
          Otherwise, $s(i,t+1) = s(i,t)$.
Step 3) Update $A(t+1)$ as follows:
          1) If $s(i,t+1) = 1$, then $[A(t+1)]_{i,j} =$
             $0$, $\forall\ 1 \leq j \leq n$.
          2) If $s(i,t+1) = 0$, then
               1) $[A(t+1)]_{i,j} = 0$, $\forall\ j$, such
                  that $\phi_j(t) = i$.
               2) $[A(t+1)]_{i,j} = [A(t)]_{i,j}/(1 -$
                  $\sum_{\alpha:\phi_\alpha(t)=i}[A(t)]_{i,\alpha})$, $\forall\ j$, such
                  that $\phi_j(t) \neq i$.
Step 4) If $t = k$ or $A(t+1) = 0$, stop. Oth-
        erwise, increase the value of $t$ by
        one and then go to step 1.
```

The conditional probability heuristic algorithm is illustrated by the following example. Suppose

$$P = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.3 & 0.25 & 0.25 & 0.2 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}.$$

Then, $A(1) = P$ and $s(1,1) = s(2,1) = s(3,1) = 0$. Since $[A(1)]_{1,1} > [A(1)]_{2,1} > [A(1)]_{3,1}$, $\phi_1(1) = 1$. Similarly, $\phi_2(1) = 1$, $\phi_3(1) = 3$ and $\phi_4(1) = 3$. Therefore, in time slot 1, the base station in cell 1 and the base station in cell 2 page mobile user 1, while the base station in cell 3 and the base station in cell 4 page mobile user 3. With probability $[A(1)]_{1,1} + [A(1)]_{1,2} = 0.4 + 0.3 = 0.7$, mobile user 1 is located in time slot 1. Similarly, with probability $[A(1)]_{3,3} + [A(1)]_{3,4} = 0.3 + 0.4 = 0.7$, mobile user 3 is located in time slot 1. For illustration purposes, it is assumed that in the first time slot, mobile user 1 is located, while mobile user 3 is not located. Therefore, $s(1,2) = 1$, $s(2,2) = 0$, and $s(3,2) = 0$.

$A(2)$ is derived based on step 3 of the algorithm. Since $s(1,2) = 1$, elements in the first row of $A(2)$ are all zeros. Since $\sum_{\alpha:\phi_\alpha(1)=2}[A(1)]_{2,\alpha} = 0$, the second row of $A(2)$ is identical to the second row of $A(1)$. Furthermore, $\sum_{\alpha:\phi_\alpha(1)=3}[A(1)]_{3,\alpha} = [A(1)]_{3,3}+[A(1)]_{3,4}= 0.3+0.4 = 0.7$ and therefore $[A(2)]_{3,1} = [A(1)]_{3,1}/(1 - 0.7) = 1/3$. Similarly, $[A(2)]_{3,2} = 0.2/0.3 = 2/3$. In addition, $[A(2)]_{3,3} = [A(2)]_{3,4} = 0$. Then

$$A(2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.3 & 0.25 & 0.25 & 0.2 \\ \frac{1}{3} & \frac{2}{3} & 0 & 0 \end{bmatrix}.$$

Next, the value of t increases from 1 to 2. Repeating the above procedures, $\phi_1(2) = 3$, $\phi_2(2) = 3$, $\phi_3(2) = 2$ and $\phi_4(2) = 2$. Then, mobile user 2 is located in time slot 2 with probability $[A(2)]_{2,3} + [A(2)]_{2,4} = 0.25 + 0.2 = 0.45$, while mobile user 3 is located in time slot 2 with probability $(1/3) + (2/3) = 1$. For illustration purposes, it is assumed that mobile user 2 is not located in time slot 2. On the other hand, since $[A(2)]_{3,1} + [A(2)]_{3,2} = 1$, mobile user 3 is located in time slot 2. Then, $s(1,3) = s(1,2) = 1$, $s(2,3) = s(2,2) = 0$, and $s(3,3) = 1$. Moreover

$$A(3) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{6}{11} & \frac{5}{11} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In time slot 3, the base station in cell 1 and the base station in cell 2 page and then locate mobile user 2. Therefore

$$A(4) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and the algorithm terminates.

B. Analysis of the Conditional Probability Heuristic Algorithm

The computational complexity of the conditional probability heuristic algorithm is derived as follows. In every time slot t, the conditional probability heuristic algorithm takes at most $(k \cdot n)$ comparisons to find $\phi_j(t)$'s, where $1 \leq j \leq n$. Furthermore, it takes at most $(k \cdot n)$ additions to calculate $\sum_{\alpha:\phi_\alpha(t)=i}[A(t)]_{i,\alpha}$, $\forall\, \phi_j(t) \neq i$. Given $\sum_{\alpha:\phi_\alpha(t)=i}[A(t)]_{i,\alpha}$, $\forall\, \phi_j(t) \neq i$ and $A(t)$, it takes $(k \cdot n)$ divisions to calculate $A(t+1)$. Since k mobile users are guaranteed to be located in k time slots, the computational complexity of the conditional probability heuristic algorithm is bounded from above by $O(k \cdot (k \cdot n + k \cdot n + k \cdot n)) = O(k^2 \cdot n)$.

VI. Numerical and Simulation Results

A. Simulation Method

The network contains $n = w^2$ cells, which are indexed by (α, β), where $0 \leq \alpha, \beta \leq w-1$. The distance between two cells indexed by (α_1, β_1) and (α_2, β_2) is denoted by $d_1(\alpha_1, \beta_1, \alpha_2, \beta_2)$ and is defined to be $|\alpha_1 - \alpha_2| + |\beta_1 - \beta_2|$. To create the tth row of a $k \times n$ probability matrix P, a cell with index (α^*, β^*) is at first randomly selected such that α^* and β^* are two independent random variables, which are uniformly distributed in $\{0, 1, 2, \ldots, w-1\}$. Then, the probability that mobile user t is in a cell indexed by (α, β) is set to be

$$\frac{r^{d_1(\alpha,\beta,\alpha^*,\beta^*)}}{\sum_{i=0}^{w-1}\sum_{j=0}^{w-1} r^{d_1(i,j,\alpha^*,\beta^*)}} \; and$$

$$[P]_{t,\alpha+\beta\cdot w+1} = \frac{r^{d_1(\alpha,\beta,\alpha^*,\beta^*)}}{\sum_{i=0}^{w-1}\sum_{j=0}^{w-1} r^{d_1(i,j,\alpha^*,\beta^*)}}$$

where the decay factor r is a real number in [0, 1]. When the decay factor is one, from the viewpoint of the system, a mobile resides equally likely in all the cells of the network. On the other hand, when the decay factor is zero, from the viewpoint of the system, a mobile user resides in a single cell with probability one. The decay factor is a parameter that determines the uniformity of the probability density function of the location of a mobile user.

The convention that the index of elements of a matrix starts from one is adopted. The two proposed concurrent search algorithms and the "brute force" search are compared to the straightforward sequential paging method that pages all cells in time slot t in order to locate mobile user t, where $1 \leq t \leq k$.

B. Simulation Results

The performances of three concurrent search algorithms are shown in Fig. 1. Due to the high computational complexity of the "brute force" algorithm, both the total number of mobile users and the total number of cells in the network are chosen to be four. When the decay factor is 0.99, mobile users appear almost equally in all the cells in the network and as shown in Fig. 1, the normalized reduction of the average paging cost due to either the simple heuristic algorithm or the conditional probability heuristic algorithm is around 25%, while that due to the "brute force" algorithm is around 37%. As the decay factor decreases, the normalized reduction of the average paging cost increases, regardless of which one of the three proposed algorithms is applied. When the decay factor is 0.01, the "brute force" algorithm, the conditional probability heuristic algorithm, and the simple heuristic algorithm yield 66%, 50%, and 37% reduction in average paging cost, respectively.

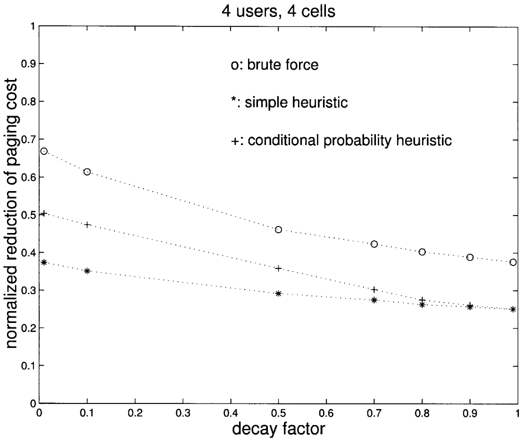

Fig. 1. Performance of concurrent search algorithms.

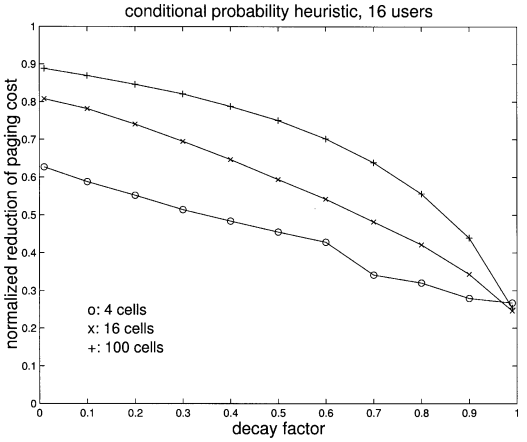

Fig. 2. Performance of the conditional probability heuristic algorithm.

Fig. 2 shows the performance of the conditional probability heuristic algorithm, when there are 16 mobile users to be located in a network composed of n cells, where $n \in \{4, 16, 100\}$. Typically, as the number of cells increases, the normalized reduction of the average paging cost increases. However, as the decay factor approaches one, the normalized reduction of the average paging cost is approximately 0.25, regardless of the total number of cells in the network. Fig. 3 shows the performance of the conditional probability heuristic algorithm, when the network is composed of 16 cells and the total number of mobile users to be located ranges from 2 to 16. It is observed that except for the case in which the decay factor is very close to one, the normalized reduction of the average paging cost increases as the total number of mobile users increases.

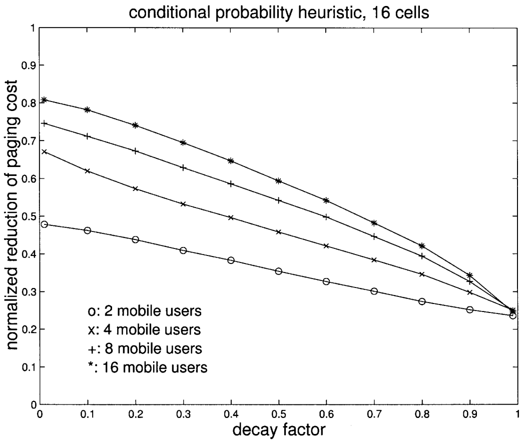

Fig. 3. Performance of the conditional probability heuristic algorithm.

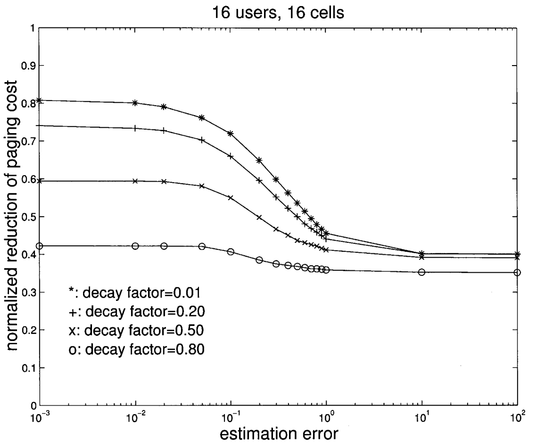

Fig. 4. Sensitivity analysis of the conditional probability heuristic algorithm.

C. Estimation Errors of Probability Density Functions

The impact of estimation errors on the performance of the conditional probability heuristic algorithm is studied in this section. To be consistent with previous notations, let $p(i, j)$ be the estimated probability that mobile user i is in cell j upon a call arrival and $q(i, j)$ be the actual probability that mobile user i is in cell j upon a call arrival. In reality, the estimated probabilities $p(i, j)$'s are not always identical to the actual probabilities $q(i, j)$'s. In our simulations, $q(i, j)$'s are derived from $p(i, j)$'s as follows. First, choose an arbitrary nonnegative real number e. Next, for every fixed pair of i and j, set $q(i, j) = p(i, j) + z(i, j)$, where $z(i, j)$'s are i.i.d. random variables that are uniformly distributed within $[-e, e]$. If $q(i, j) > 1$, set $q(i, j) = 1$. Similarly, if $q(i, j) < 0$, set $q(i, j) = 0$. At last, for every fixed i, normalize $q(i, j)$ such that $\sum_{j=1}^{n} q(i, j) = 1$. In our simulations, updated $p(i, j)$'s that form $A(t)$'s are used by the conditional probability heuristic algorithm to make paging decisions, while updated $q(i, j)$'s are used to determine if a mobile user is in a specific cell in a time slot. The updating procedure of $q(i, j)$'s is very similar to that of $p(i, j)$'s.

Fig. 4 shows the impact of estimation errors on the performance of the conditional probability heuristic algorithm, when there are 16 mobile users to be located in a network constituting of 16 cells. The variable in the *x*-axis is called the estimation error and is equivalent to the value of the variable e in the previous paragraph. The decay factor used to generate $p(i, j)$'s ranges from 0.01 to 0.8. When the decay factor is 0.01 and the estimation error e is no greater than 0.001, the normalized reduction of the average paging cost is 80.8%. As the estimation error goes up to 0.1, the normalized reduction of the average paging cost reduces to 72%. When the estimation error is greater than 0.1, the normalized reduction of the average paging cost significantly decreases. On the other hand, when the decay factor is 0.8, as the estimation error increases from 0 to 0.1, the normalized reduction of the average paging cost slightly decreases from 42.2% to 40.7%. It is observed that when the decay factor is smaller, the conditional probability heuristic algorithm has a better performance but is more vulnerable to estimation errors.

VII. Discussions

A. Multiple Paging Channels, Multiple Batches of Call Arrivals

In this section, the conditional probability heuristic algorithm is modified to concurrently search for mobile users when there are multiple paging channels and the arrival times of calls to mobile users are arbitrary. As in most of the slotted systems, in order to be served, a call that arrives in the middle of a time slot has to wait till the beginning of the next time slot. Therefore, without loss of essential generality, it is assumed that every call to a mobile user arrives in the beginning of a time slot.

We first define some terms for the modified conditional probability heuristic algorithm. Let m be the total number of available paging channels. Recall that n is the total number of cells in the network. In the control center of the network, n queues are created and each queue corresponds to a cell in the network. Denote these n queues by $Q_1, Q_2, \ldots, Q_n$. Let t be the current time in unit of time slots. Let $Y(t)$ be the number of calls arrive at the beginning of time slot t. Without loss of essential generality, it is assumed that the ith mobile user is the callee of the ith call. When the ith call arrives, the control center creates n packets, denoted by $PK_1(i), PK_2(i), \ldots, PK_n(i)$. The primary index of $PK_j(i)$, $\forall\ 1 \leq j \leq n$, is i. Let $A_j(i)$ be the arrival time of $PK_j(i)$ and $V_j(i)$ be the value of $PK_j(i)$. For each fixed value of i, $A_\alpha(i) = A_\beta(i)$, $\forall\ 1 \leq \alpha, \beta \leq n$, since both $PK_\alpha(i)$ and $PK_\beta(i)$ are associated with the ith call. $V_j(i)$ is the conditional probability that the ith mobile user is in cell j. It should be emphasized that the value of $V_j(i)$ may change with time. Let $\lambda > 1$ be an arbitrary real number. Let $PR_j(i) = \lambda \cdot A_j(i) - V_j(i)$ be the priority of the packet $PK_j(i)$. Mobile user i is said to be unlocated, if the ith call has arrived but the ith mobile user has not been located by the system. Let Θ be the list that contains all the indexes of unlocated mobile users.

The modified conditional probability heuristic algorithm is shown as follows:

```
Step 0) Initially, all $n$ queues in the
        control center are empty. Set $t = 1$
        and $i = 1$. Choose $\lambda > 1$. The unlo-
        cated mobile user list $\Theta$ is empty.
Step 1) Let $Y(t)$ be the number of calls
        arrive at the beginning of time
        slot $t$. For each of these $Y(t)$
        calls, sequentially do the following and then go to step 2.
          1) Add the index $i$ to the list $\Theta$.
          2) For each queue $j$, where $j \in \{1,2,3,\ldots,n\}$, create $PK_j(i)$, set $A_j(i) = t$, let $V_j(i)$ be the probability that mobile user $i$ is in cell $j$, set $PR_j(i) = \lambda \cdot A_j(i) - V_j(i)$, and put $PK_j(i)$ into queue $j$.
          3) Increase the value of $i$ by one.
Step 2) For each value of $j$, where $j \in \{1,2,3,\ldots,n\}$, do the following.
          1) Select and remove $s(j) = \min\{m, q_j(t)\}$ packets with the highest priorities from the $j$th queue, where $q_j(t)$ is the total number of packets in queue $j$ at time slot $t$.
          2) Let $f(j,1), f(j,2), \ldots, f(j,s(j))$ be integers such that $PK_j(f(j,1)), PK_j(f(j,2)), \ldots, PK_j(f(j,s(j)))$ are the above selected packets. Command base station $j$ to page mobile users $f(j,1), f(j,2), \ldots, f(j,s(j))$.
Step 3) Wait for responses from all the base stations till the end of the current time slot.
Step 4) Let $L(t)$ be the total number of mobile users that are located in time slot $t$. In addition, let $g(1), g(2), \ldots, g(L(t))$ be the indexes of the mobile users that are located in time slot $t$. For each value of $\mu$, where $\mu \in \{g(1), g(2), \ldots, g(L(t))\}$, do the following.
          1) For each value of $j$, where $1 \le j \le n$, if $PK_j(\mu) \in Q_j$, then remove $PK_j(\mu)$ from $Q_j$.
          2) Remove $\mu$ from the unlocated mobile user list $\Theta$.
Step 5) For each mobile user $\mu$ that is paged but not located in the current time slot, do the following.
          1) $x_\mu = \sum_{j=1}^{n} V_j(\mu) \cdot I(PK_j(\mu) \in Q_j)$
          2) $V_j(\mu) \leftarrow V_j(\mu) \cdot x_\mu^{-1}$, $\forall\ j$, where $1 \le j \le n$ and $PK_j(\mu) \in Q_j$
          3) $PR_j(\mu) = \lambda \cdot A_j(\mu) - V_j(\mu)$, $\forall\ j$, where $1 \le j \le n$ and $PK_j(\mu) \in Q_j$
Step 6) Increase the value of $t$ by one and then go to step 1.
```

The above algorithm is explained as follows. Step 0 is the initialization of the algorithm. All n queues in the control center are initially empty. The index of a time slot starts from 1. Similarly, the index of a call starts from 1. Furthermore, there are no unlocated mobile users, since no calls have arrived yet.

In step 1, the control center starts to process the new arriving calls. For each new arriving call, indexed by i, the control center do the following. First, the control center adds the index of the call into the unlocated mobile user list Θ. This means that the system has to search for mobile user i. Second, the control center creates n packets, denoted by $PK_1(i), PK_2(i), \ldots, PK_n(i)$. For each value of j, where $1 \le j \le n$, the arrival time of $PK_j(i)$ is set to be the current time t, while the value of $PK_j(i)$ is set to be the initial estimation of the probability that mobile user i is in cell j. The priority of a packet is a function of the arrival time of the packet and the current value of the packet. In particular, $PR_j(i) = \lambda \cdot A_j(i) - V_j(i)$. Then, the control center inserts the packet $PK_j(i)$ into queue j, $\forall\, 1 \le j \le n$.

In step 2, the control center selects the mobile users to be paged by each of the n base stations. For each base station j, the control center selects $s(j) = \min\{m, q_j(t)\}$ packets from Q_j, according to the priorities of packets. The parameter m is the total number of paging channels, while $q_j(t)$ is the total number packets in queue j at time slot t. If a packet with primary index i is selected from Q_j, base station j will page mobile user i in the current time slot. Once a packet is selected, it is immediately removed from the queue to avoid paging a mobile user in a cell twice. In the algorithm, the integers $f(j,1), f(j,2), \ldots, f(j,s(j))$ are the indexes of mobile users that will be paged by base station j at time t.

In step 3, the system simply waits for the responses from the paged mobile users till the end of the current time slot.

In step 4, since $L(t)$ mobile users are located in the current time slot, the control center updates $Q_1, Q_2, \ldots, Q_n$ and the unlocated mobile user list Θ. The variables $g(1), g(2), \ldots, g(L(t))$ represent the indexes of mobile users that are located in the current time slot. For each mobile user μ that is located in the current time slot, every packet with primary index equivalent to μ is removed from the associated queue and discarded. Due to the removal operations, it is guaranteed that a base station will not page a mobile user that has been located in some other cell. In addition, since mobile user μ has been located, the index μ is removed from the unlocated mobile user list Θ.

In step 5, the control center updates the values and priorities of packets associated with mobile users that are paged in the current time slot but has not yet been located. For each such mobile user μ, the value of x_μ is set to be $\sum_{j=1}^{n} V_j(\mu) \cdot I(PK_j(\mu) \in Q_j)$, where $I(\cdot)$ is the indicator function with value one if the condition inside is true or with value zero otherwise. Therefore, x_μ is the summation of the values of $PK_j(\mu)$'s, where $1 \le j \le n$ and $PK_j(\mu)$ is in queue j. Then, based on the definition of the conditional probability, the value of each packet $PK_j(\mu)$, which remains in queue j, is updated by dividing itself by x_μ. Furthermore, the priority of the packet $PK_j(\mu)$ is also updated. More precisely, $PR_j(\mu) = \lambda \cdot A_j(\mu) - V_j(\mu)$, $\forall\, j$, where $1 \le j \le n$ and $PK_j(\mu) \in Q_j$.

In step 6, the index of the current time slot is increased by one and then the above steps are repeated to process new arriving calls as well as the old calls with callees that have not been located by the system.

The modified conditional probability heuristic algorithm is further illustrated by the following example. Suppose there are

$n = 3$ cells in the network and there are $m = 2$ paging channels in each cell. The parameter λ is chosen to be 2. It is assumed that three calls arrive at the beginning of time slot 1, while four calls arrive at the beginning of time slot 2. Therefore, $\forall\, j \in \{1, 2, 3\}$, $A_j(1) = A_j(2) = A_j(3) = 1$ and $A_j(4) = A_j(5) = A_j(6) = A_j(7) = 2$. Suppose that from the viewpoint of the system, mobile user 1 is in cell 2 with probability 0.4 at the beginning of time slot 1. Thus, $V_2(1) = 0.4$. Similarly, it is assumed that $V_1(1) = 0.4$, $V_3(1) = 0.2$, $V_1(2) = 0.3$, $V_2(2) = 0.3$, $V_3(2) = 0.4$, $V_1(3) = 0.2$, $V_2(3) = 0.7$, $V_3(3) = 0.1$ at the beginning of time slot 1. Since $PR_j(i) = \lambda \cdot A_j(i) - V_j(i)$, $PR_1(1) = 2 \cdot 1 - 0.4 = 1.6$, $PR_1(2) = 2 \cdot 1 - 0.3 = 1.7$ and $PR_1(3) = 2 \cdot 1 - 0.2 = 1.8$. Therefore, in time slot 1, the control center selects $PK_1(1)$ and $PK_1(2)$ from Q_1, and then base station 1 pages mobile user 1 and mobile user 2. Similarly, base station 2 pages mobile user 1 and mobile user 3, while base station 3 pages mobile user 1 and mobile user 2. For illustration purposes, it is assumed that mobile user 1 resides in cell 2, mobile user 2 resides in cell 3, and mobile user 3 resides in cell 1. Therefore, in the end of time slot 1, both mobile user 1 and mobile user 2 have been located, but mobile user 3 has not yet been located. Next, according to the modified condition probability heuristic algorithm, all packets associated with either mobile user 1 or mobile user 2 are removed from the queues in the control center. On the other hand, the values of packets associated with mobile user 3 are updated. In particular, $V_1(3) = 0.2/(0.2 + 0.1) = 2/3$, and $V_3(3) = 0.1/(0.2 + 0.1) = 1/3$.

In the beginning of time slot 2, four calls arrive. In addition, it is assumed that $V_1(4) = 0.8$, $V_2(4) = 0.1$, $V_3(4) = 0.1$, $V_1(5) = 0.7$, $V_2(5) = 0.2$, $V_3(5) = 0.1$, $V_1(6) = 0.1$, $V_2(6) = 0.5$, $V_3(6) = 0.4$, $V_1(7) = 0.2$, $V_2(7) = 0.3$, $V_3(7) = 0.5$. Although $V_1(4) > V_1(5) > V_1(3)$, due to that $\lambda > 1$ and $A_1(3) < A_1(4) = A_1(5)$, $PR_1(3) = 2 \cdot 1 - (2/3) = (4/3) < PR_1(4) = 2 \cdot 2 - 0.8 = 3.2 < PR_1(5) = 2 \cdot 2 - 0.7 = 3.4$. Therefore, at the beginning of time slot 2, the control center selects $PK_1(3)$ and $PK_1(4)$ from Q_1, and then base station 1 pages mobile user 3 and mobile user 4 simultaneously. Similarly, base station 3 pages mobile user 3 and mobile user 7. In this example, in the beginning of time slot 2, the corresponding call arrival time of mobile user 3 is smaller than that of any other unlocated mobile user. As a result, each remaining packet associated with mobile user 3 is selected, which in turn assures that mobile user 3 will be located in the end of time slot 2.

We are currently working on the performance evaluation of the modified conditional probability heuristic algorithm. The above sketch of the algorithm is used to demonstrate that the concurrent search approach is still applicable when there are multiple paging channels and the arrival times of calls are arbitrary.

B. Suboptimality of the Conditional Probability Heuristic Algorithm

It should be noted that the conditional probability heuristic algorithm does not always produce an optimal solution. Instead, the concept of $PR_j(i)$ is introduced to simultaneously reduce the average paging cost and provide a relative bound of paging delay. Since the call arrival times are random in general, the existence of queueing delay makes it very difficult to locate a mobile user within a nontrivial delay bound. However, with our approach, it is guaranteed that regardless of the probabilistic location information, a mobile user will be located in the current time slot, if all mobile users with earlier or equivalent call arrival times had been located. Finding better concurrent search algorithms is a possible direction of future research.

C. Second-Generation Cellular Networks and UMTS

To simplify the explanation, in the beginning of this paper, it is assumed that a cellular network is composed of a single location area. A typical cellular network is composed of a number of location areas and the proposed concurrent search approach could be used to search for mobile users in a location area. Next generation mobile systems, such as UMTS and IMT-2000, will be multitier cellular networks [35]. The concurrent search approach remains applicable there.

VIII. Summary and Conclusion

We have proposed the concurrent search approach to locate a batch of k mobile users in the cellular network within k time slots based on the probabilistic information about the locations of mobile users. We have shown that even when there is only one paging channel in a cell, based on the concurrent search approach, it is possible to simultaneously locate a number of mobile users and therefore reduce the average paging cost.

We have proposed three concurrent search algorithms: the "brute force" algorithm, the simple heuristic algorithm, and the conditional probability heuristic algorithm. Although the "brute force" algorithm outperforms the other two algorithms, it is not feasible in practice for a large-scale network due to its extremely high computational complexity. It has been shown that regardless of the probabilistic information about the locations of mobile users, the simple heuristic algorithm asymptotically guarantees 25% reduction in the average paging cost. This result provides an analytical support for the concurrent search approach. We have also studied the performance of the proposed algorithms by simulations. Due to its high computational complexity, the "brute force" algorithm can be used only when the total number of mobile users to be located and the total number of cells in the network are both very small. In this case, simulation results have indicated that the performance difference between the conditional probability heuristic algorithm and the "brute force" algorithm is within 20%. While the simple heuristic algorithm is inherently unable to yield more than 50% reduction in the average paging cost, the conditional probability heuristic algorithm is able to produce more than 50% reduction in the average paging cost. Except for the cases in which every mobile user appears almost equally in all the cells of the network, the conditional probability heuristic algorithm is always superior to the simple heuristic algorithm. As the total number of mobile users increases, the performance difference between the conditional probability heuristic algorithm and the simple heuristic algorithm increases. Depending on the total number of cells in the network, total number of mobile users to be located, and the probabilistic information about the locations of mobile users, the reduction of the average paging cost due to the conditional probability heuristic algorithm ranges from 25% to 88%. As the total number of mobile users to be located increases, the performance of the conditional probability heuristic algorithm becomes better. Similarly, as the total number of cells in the network becomes larger, the conditional probability heuristic algorithm produces more reduction in the average paging cost. As the decay factor approaches zero, a mobile user tends to appear in a very small set of cells and it takes minimum paging cost to locate the mobile user.

As the cell size in the next-generation cellular network becomes smaller, the total number of cells in the network increases. Furthermore, the total number of mobile users will continue to increase as well. Therefore, being able to locate mobile users with minimum average paging cost is an important benefit. Our proposed concurrent search scheme could be applied in the mobility management protocols in the next-generation cellular network. Among three proposed algorithms, we recommend the conditional probability heuristic algorithm that performs well and does not require high computational complexity.

Appendix

Lemma 1: When k divides n, an optimal schedule is $x(i,j) = i - 1 + \lceil (j \cdot k)/n \rceil - k \cdot I(i - 1 + \lceil (j \cdot k)/n \rceil > k)$, $\forall\, 1 \le i \le k, 1 \le j \le n$.

Proof:

1) Since $1 \le i \le k$ and $1 \le j \le n$, $0 + 1 \le i - 1 + \lceil (j \cdot k)/n \rceil \le k - 1 + k$. Let $z(i,j) = i - 1 + \lceil (j \cdot k)/n \rceil$. Then, $1 \le z(i,j) \le 2k - 1$. When $1 \le z(i,j) \le k$, $z(i,j) - k \cdot I(z(i,j) > k) = z(i,j)$ and therefore $1 \le z(i,j) - k \cdot I(z(i,j) > k) \le k$. On the other hand, when $k + 1 \le z(i,j) \le 2k - 1$, $k + 1 - k \le z(i,j) - k \cdot I(z(i,j) > k) \le 2k - 1 - k$. Therefore, $1 \le x(i,j) \le k$.
2) $\forall\, j$, if $i_1 \ne i_2$

$$x(i_1,j) - x(i_2,j) = i_1 - i_2 + k \cdot \left(I\left(z(i_1,j) > k\right) - I\left(z(i_2,j) > k\right)\right).$$

Since $1 \le i_1, i_2 \le k$ and $i_1 \ne i_2$, $0 < |i_1 - i_2| < k$. Then

$$x(i_1,j) - x(i_2,j) \equiv i_1 - i_2 \pmod{k}$$

Since $i_1 \ne i_2$, $x(i_1,j) - x(i_2,j) \ne 0$.

3) From 1 and 2, $x(i,j)$, where $1 \le i \le k$, $1 \le j \le n$, is a feasible schedule. Furthermore, the corresponding $m(i,s)$'s, where $1 \le i, s \le k$, are all equivalent to n/k. Hence, $x(i,j) = i - 1 + \lceil (j \cdot k)/n \rceil - k \cdot I(i - 1 + \lceil (j \cdot k)/n \rceil > k)$, where $1 \le i \le k$, $1 \le j \le n$, forms an optimal schedule.

Lemma 2: When k divides n, $r^*(k, n, P_{U(k,n)}) = (1/2) - (1/(2 \cdot k))$.

Proof:

1) When k divides n, an optimal schedule is shown in the above lemma. Therefore

$$\begin{aligned} C_k^*\left(P_{U(k,n)}\right) &= k \cdot \sum_{s=1}^{k} \left(\frac{1}{k}\right) \cdot \left(\frac{n \cdot s}{k}\right) \text{ (symmetry)} \\ &= \frac{(k+1) \cdot n}{2}. \end{aligned}$$

2) Then

$$r^*\left(k, n, P_{U(k,n)}\right) = \frac{k \cdot n - \frac{(k+1) \cdot n}{2}}{k \cdot n} = \frac{1}{2} - \frac{1}{2 \cdot k}.$$

Lemma 4: $C_2^*(P_{U(2,n)}) \ge C_2^*(P)$, where n is an even number and P is an arbitrary $2 \times n$ probability matrix.

Proof:

1) In this case, $k = 2$. Recall that $m(i,s)$, where $1 \le i$, $s \le k$, is the cardinality of the sth paging zone of mobile user i. Moreover, m is the collection of $m(i,s)$'s, where $1 \le i, s \le k$. Let $C(m)$ be the average paging cost, when the probability matrix is $P_{U(2,n)}$ and the schedule is m. Then

$$\begin{aligned} C(m) &= \sum_{i=1}^{2} \left\{ \frac{m(i,1)}{n} \cdot m(i,1) + \frac{m(i,2)}{n} \cdot (m(i,1) + m(i,2)) \right\} \\ &= \frac{1}{n} \sum_{i=1}^{2} \left[m(i,1)^2 + (n - m(i,1)) \cdot n \right] \\ &= \frac{3n}{2} + \frac{2}{n} \cdot \left(m(1,1) - \frac{n}{2} \right)^2. \end{aligned}$$

Clearly, $C(m)$ is a convex function of $m(1,1)$. When n is even, $C(m)$ is minimized at $m(1,1) = n/2$. Then, $C_2^*(P_{U(2,n)}) = \min_m C(m) = 3n/2$.

2) By definition, $C_2^*(P) \le C_2(P, X_s)$. When n is even, $C_2(P, X_s) \le 3n/2$. Then

$$\begin{aligned} C_2^*(P) &\le C_2(P, X_s) \\ &\le \frac{3n}{2} \\ &= C_2^*\left(P_{U(2,n)}\right). \end{aligned}$$

Lemma 5: If $n = k = 2$ and $p(1,j) = p(2,j), \forall\, 1 \le j \le n$, then $C_2^*(P) = C_2^*(P_{U(2,n)})$.

Proof:

1) It is clear that $C_2^*(P_{U(2,2)}) = (3 \cdot 2)/2 = 3$. Let X_i, where $1 \le i \le 4$, be a schedule. Let

$$X_1 = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}, X_2 = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix}, X_3 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, X_4 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

2) $C(P, X_1) = 2 + 2 = 4$. $C(P, X_2) = 2 + 2 = 4$.

$$\begin{aligned} C(P, X_3) &= [p(1,1) + 2 \cdot p(1,2)] + [p(2,2) + 2 \cdot p(2,1)] \\ &= 2 + [p(2,2) + p(2,1)], \ (p(1,2) = p(2,2)) \\ &= 3. \end{aligned}$$

Similarly, $C(P, X_4) = 3$.

3) Since X_1, X_2, X_3, X_4 are the only valid schedules and $\min_{i \in \{1,2,3,4\}} C(P, X_i) = 3$, $C_2^*(P) = C_2^*(P_{U(2,2)})$.

Lemma 6: If k divides n and P_I is a $k \times n$ row-identical probability matrix, $C_k^*(P_I) \leq C_k^*(P_{U(k,n)})$.

Proof:

1) Choose a schedule X such that $x(i,j) = i - 1 + \lceil (j \cdot k)/n \rceil - k \cdot I(i-1+\lceil (j \cdot k)/n \rceil > k)$, where $1 \leq i \leq k$, $1 \leq j \leq n$. Clearly, $m(i,s) = n/k, \forall\ 1 \leq i, s \leq k$. Furthermore, $\pi(i,s) = \pi(1, s-i+1+k \cdot I(s-i+1 \leq 0))$, where $1 \leq i, s \leq k$.
2) Let $\phi_s = \pi(1,s)$, where $1 \leq s \leq k$. Note that $\sum_{i=1}^{k} \phi_i = 1$. Then

$$\begin{aligned} C(P_I, X) &= \sum_{i=1}^{k} \sum_{s=1}^{k} \pi(i,s) \cdot \frac{n \cdot s}{k} \\ &= \left(\sum_{i=1}^{k} \phi_i \right) \cdot \left(\frac{n}{k} \right) + \left(\sum_{i=1}^{k} \phi_i \right) \\ &\quad \cdot \left(\frac{2n}{k} \right) + \ldots + \left(\sum_{i=1}^{k} \phi_i \right) \cdot \left(\frac{kn}{k} \right) \\ &= \frac{(k+1)n}{2}. \end{aligned}$$

3) Recall that when k divides n, $C_k^*(P_{U(k,n)}) = ((k+1)n)/2$. Then

$$\begin{aligned} C_k^*(P_I) &\leq C(P_I, X) \\ &= \frac{(k+1)n}{2} \\ &= C_k^*\left(P_{U(k,n)}\right). \end{aligned}$$

Lemma 7: $D_2^*(P_{U(2,n)}) \geq D_2^*(P)$, where P is an arbitrary $2 \times n$ probability matrix.

Proof:

1) Let X_s be the schedule obtained by running the simple heuristic algorithm with input P. When $\phi_1 \geq \phi_2$

$$\begin{aligned} D(P, X_s) &= \frac{1}{2} \{ [\phi_1 \cdot 1 + (1-\phi_1) \cdot 2] + [(1-\phi_2) \cdot 1 + \phi_2 \cdot 2] \} \\ &= \frac{3}{2} - \frac{\phi_1 - \phi_2}{2} \\ &\leq \frac{3}{2}. \end{aligned}$$

When $\phi_1 < \phi_2$, it can be similarly proved that $D(P, X_s) \leq 3/2$. Therefore, $D(P, X_s) \leq 3/2$.

2) Consider the case when the probability matrix is $P_{U(2,n)}$. Let $k = 2$. Recall that $m(i,s) = |Z(i,s)|$, where $1 \leq i$, $s \leq k$, is the total number of cells in the sth paging zone of mobile user i. Let $m = \{m(i,s) | 1 \leq i, s \leq k\}$ and $D(m)$ be the average paging delay associated with m. Then

$$\begin{aligned} D(m) &= \frac{1}{2n} \cdot \{ [m(1,1) \cdot 1 + m(1,2) \cdot 2] \\ &\qquad + [m(2,1) \cdot 1 + m(2,2) \cdot 2] \} \\ &= \frac{1}{2n} \cdot \{ [m(1,1) + (n - m(1,1)) \cdot 2] \\ &\qquad + [(n - m(1,1)) \cdot 1 + m(1,1) \cdot 2] \} \\ &= \frac{3}{2}. \end{aligned}$$

Thus, $D_2^*(P_{U(2,n)}) = \min_m D(m) = 3/2$.

3) Then

$$\begin{aligned} D_2^*\left(P_{U(2,n)}\right) &= \frac{3}{2} \\ &\geq D(P, X_s) \\ &\geq D_2^*(P). \end{aligned}$$

Lemma 8: If $k = 2$ and $p(1,j) = p(2,j), \forall\ 1 \leq j \leq n$, then $D_2^*(P) = D_2^*(P_{U(2,n)}), \forall\ n \geq 2$.

Proof:

1) Since $k = 2$, for any valid schedule X, $\forall\ 1 \leq j \leq n$, $x(1,j) = 1$ implies $x(2,j) = 2$ and vice versa. Similarly, $\forall\ 1 \leq j \leq n$, $x(1,j) = 2$ implies $x(2,j) = 1$ and vice versa. Then, for any valid schedule X

$$\begin{aligned} \pi(1,1) &= \sum_{j=1}^{n} I\left(x(1,j) = 1\right) \cdot p(1,j) \\ &= \sum_{j=1}^{n} I\left(x(2,j) = 2\right) \cdot p(2,j) \\ &= \pi(2,2). \end{aligned}$$

Similarly, it can be proved that $\pi(1,2) = \pi(2,1)$.

2) It is straightforward to show that $D(P,X) = 3/2$. Therefore, $D_2^*(P) = \min_X D(P,X) = 3/2 = D_2^*(P_{U(2,n)})$.

Lemma 9: If k divides n and P_I is a $k \times n$ row-identical probability matrix, then $D_k^*(P_I) \leq D_k^*(P_{U(k,n)})$.

Proof:

1) Choose a schedule X such that $x(i,j) = i - 1 + \lceil (j \cdot k)/n \rceil - k \cdot I(i-1+\lceil (j \cdot k)/n \rceil > k)$, where $1 \leq i \leq k$, $1 \leq j \leq n$. Clearly, $m(i,s) = n/k, \forall\ 1 \leq i, s \leq k$. Furthermore, $\pi(i,s) = \pi(1, s-i+1+k \cdot I(s-i+1 \leq 0))$, where $1 \leq i, s \leq k$.
2) Let $\phi_s = \pi(1,s)$, where $1 \leq s \leq k$. Then

$$\begin{aligned} D(P_I, X) &= \frac{1}{k} \sum_{i=1}^{k} \sum_{s=1}^{k} \pi(i,s) \cdot s \\ &= \frac{1}{k} \left[\left(\sum_{i=1}^{k} \phi_i \right) \cdot 1 + \left(\sum_{i=1}^{k} \phi_i \right) \cdot 2 \right. \\ &\qquad \left. + \ldots + \left(\sum_{i=1}^{k} \phi_i \right) \cdot k \right] \\ &= \frac{(k+1)}{2}. \end{aligned}$$

3) Recall that when k divides n, $D_k^*(P_{U(k,n)}) = (k+1)/2$. Then

$$\begin{aligned} D_k^*(P_I) &\leq D(P_I, X) \\ &= \frac{(k+1)}{2} \\ &= D_k^*\left(P_{U(k,n)}\right). \end{aligned}$$

References

[1] D. Munoz-Rodriguez, "Cluster paging for traveling subscribers," in *Proc. 40th IEEE Vehicular Technology Conf.*, May 1990, pp. 748–753.

[2] H. Xie, S. Tabbane, and D. Goodman, "Dynamic location area management and performance analysis," in *Proc. 43rd IEEE Vehicular Technology Conf.*, May 1993, pp. 535–539.

[3] A. Bar-Noy and I. Kessler, "Tracking mobile users in wireless communication networks," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1877–1886, Nov. 1993.

[4] D. Plassmann, "Location management strategies for mobile cellular networks of 3rd generation," in *Proc. 44th IEEE Vehicular Technology Conf.*, June 1994, pp. 649–653.

[5] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?," *Wireless Networks*, vol. 1, no. 2, pp. 175–185, July 1995.

[6] C. Rose and R. Yates, "Minimizing the average cost of paging under delay constraints," *Wireless Networks*, vol. 1, no. 2, pp. 211–219, Feb. 1995.

[7] U. Madhow, M. L. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking," *IEEE/ACM Trans. Networking*, vol. 3, pp. 698–707, Dec. 1995.

[8] J. S. M. Ho and I. F. Akyildiz, "Mobile user location update and paging under delay constraints," *Wireless Networks*, vol. 1, no. 4, pp. 413–425, Dec. 1995.

[9] I. F. Akyildiz and J. S. M. Ho, "Dynamic mobile user-location update for wireless PCS networks," *Wireless Networks*, vol. 1, no. 2, pp. 187–196, July 1995.

[10] G. L. Lyberopoulos, J. G. Markoulidakis, D. V. Polymeros, D. F. Tsirkas, and E. D. Sykas, "Intelligent paging strategies for third generation mobile telecommunication systems," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 543–553, Aug. 1995.

[11] Y. Birk and Y. Nachman, "Using direction and elapsed-time information to reduce the wireless cost of locating mobile units in cellular networks," *Wireless Networks*, vol. 1, no. 4, pp. 403–412, Dec. 1995.

[12] C. Rose and R. Yates, "Minimizing the average cost of paging and registration: a timer-based method," *Wireless Networks*, vol. 2, no. 2, pp. 109–116, June 1996.

[13] I. F. Akyildiz, J. S. M. Ho, and Y.-B. Lin, "Movement-based location update and selective paging for PCS networks," *IEEE/ACM Trans. Networking*, vol. 4, pp. 629–638, Aug. 1996.

[14] S. J. Kim and C. Y. Lee, "Modeling and analysis of the dynamic location registration and paging in microcellular systems," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 82–90, Feb. 1996.

[15] G. Wan and E. Lin, "A dynamic paging scheme for wireless communication systems," in *Proc. MOBICOM 1997*, pp. 195–203.

[16] A. Abutaleb and V. O. K. Li, "Paging strategy optimization in personal communication systems," *Wireless Networks*, vol. 3, no. 3, pp. 195–204, 1997.

[17] ——, "Location update optimization in personal communication systems," *Wireless Networks*, vol. 3, no. 3, pp. 205–216, 1997.

[18] S. K. Das and S. K. Sen, "A new location update strategy for cellular networks and its implementation using a genetic algorithm," in *Proc. MOBICOM 1997*, pp. 185–194.

[19] T. X. Brown and S. Mohan, "Mobility management for personal communication systems," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 269–278, May 1997.

[20] A. Hac and X. Zhou, "Location strategies for personal communication networks: a novel tracking strategy," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1425–1436, Oct. 1997.

[21] G. P. Pollini and C.-L. I, "A profile-based location strategy and its performance," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1415–1424, Oct. 1997.

[22] C. Rose, "State-based paging/registration: a greedy technique," *IEEE Trans. Veh. Technol.*, vol. 48, pp. 166–173, Jan. 1999.

[23] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for PCS networks," in *Proc. IEEE INFOCOM 1999*, pp. 1377–1384.

[24] A. Bhattacharya and S. K. Das, "LeZi-update: an information-theoretic approach to track mobile users in PCS networks," in *Proc. MOBICOM 1999*, pp. 1–12.

[25] B. Krishnamachari, R.-H. Gau, S. B. Wicker, and Z. J. Haas, "Optimal sequential paging in cellular networks," in *Proc. 54th IEEE Vehicular Technology Conf. (Fall 2001)*, vol. 1, Oct. 2001, pp. 92–96.

[26] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 2nd ed. New York: Wiley, 1985.

[27] W. H. A. Yuen and W. S. Wong, "A contention-free mobility management scheme based on probabilistic paging," *IEEE Trans. Veh. Technol.*, vol. 50, pp. 48–58, Jan. 2001.

[28] W. S. Jeon and D. G. Jeong, "Performance of improved probabilistic location update scheme for cellular mobile networks," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 2164–2173, Nov. 2000.

[29] C. U. Saraydar, O. E. Kelly, and C. Rose, "One-dimensional location area design," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1626–1632, Sept. 2000.

[30] L. P. Araujo and J. R. B. de Marca, "Paging and location update algorithms for cellular systems," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 1606–1614, Sept. 2000.

[31] V. W.-S. Wong and V. C. M. Leung, "Location management for next-generation personal communications networks," *IEEE Network*, vol. 14, pp. 18–24, Sept.–Oct. 2000.

[32] J. Li, H. Kameda, and K. Li, "Optimal dynamic mobility management for PCS networks," *IEEE/ACM Trans. Networking*, vol. 8, pp. 319–327, June 2000.

[33] R. Rezaiifar and A. Makowski, "From optimal search theory to sequential paging in cellular networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1253–1264, Sept. 1997.

[34] W. Wang, I. Akyildiz, and G. Stuber, "Effective paging schemes with delay bounds as QoS constraints in wireless systems," *Wireless Networks*, vol. 7, pp. 455–466, Sept. 2001.

[35] M. Madfors, K. Wallstedt, S. Magnusson, H. Olofsson, P.-O. Backman, and S. Engstrom, "High capacity with limited spectrum in cellular systems," *IEEE Commun. Mag.*, vol. 35, pp. 38–45, Aug. 1997.

Rung-Hung Gau received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1994, the M.S.E.E. degree from the University of California at Los Angeles in 1997, and the Ph.D. degree from Cornell University, Ithaca, NY, in 2001.

He joined the faculty of the Institute of Communications Engineering, National Sun Yat-Sen University, Taiwan, in February 2002. His research interests include mobility and resource management in wireless and mobile networks, multicast flow and congestion control, probabilistic models and optimization techniques for network problems.

Zygmunt J. Haas (S'84–M'88–SM'90) received the B.Sc. degree in electrical engineering in 1979 and the M.Sc. degree in electrical engineering in 1985. He received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1988.

He joined AT&T Bell Laboratories Network Research Department in 1988, where he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 to July 1995, he was with the AT&T Wireless Center of Excellence, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. He joined the faculty of the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, in August 1995. He is an author of numerous technical papers and holds 15 patents in the fields of high-speed networking, wireless networks, and optical switching. His research interests include mobile and wireless communication and networks, personal communication service, and high-speed communication and protocols.

Dr. Haas has organized several workshops, delivered numerous tutorials at major IEEE and ACM conferences, and serves as editor of several journals and magazines, including the IEEE TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE *Communications Magazine*, and the ACM/Kluwer *Wireless Networks*. He has been a guest editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS issues on Gigabit Networks, Mobile Computing Networks, and Ad-Hoc Networks. He is a voting member of the Association for Computing Machinery and the Chair of the IEEE Technical Committee on Personal Communications.